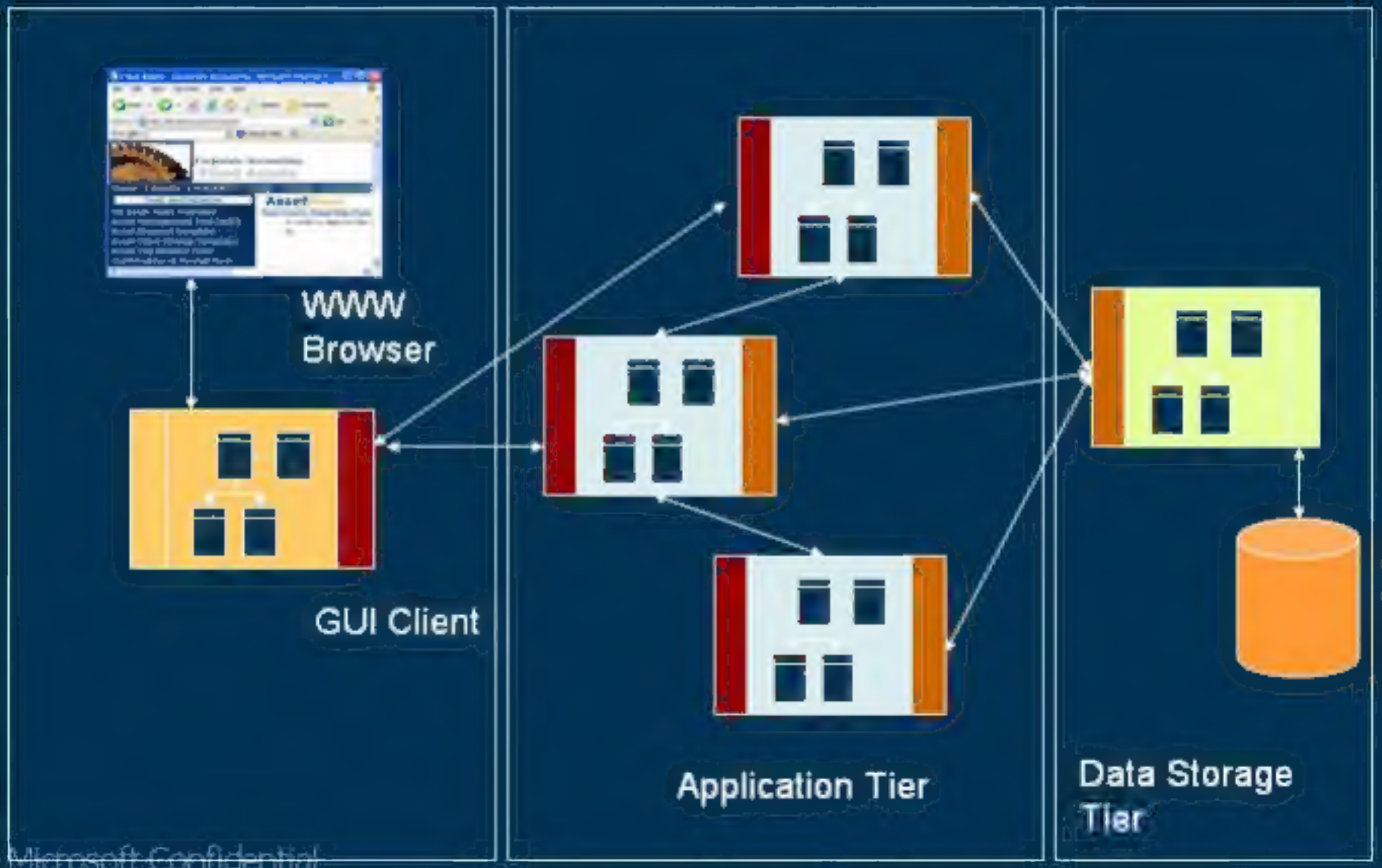
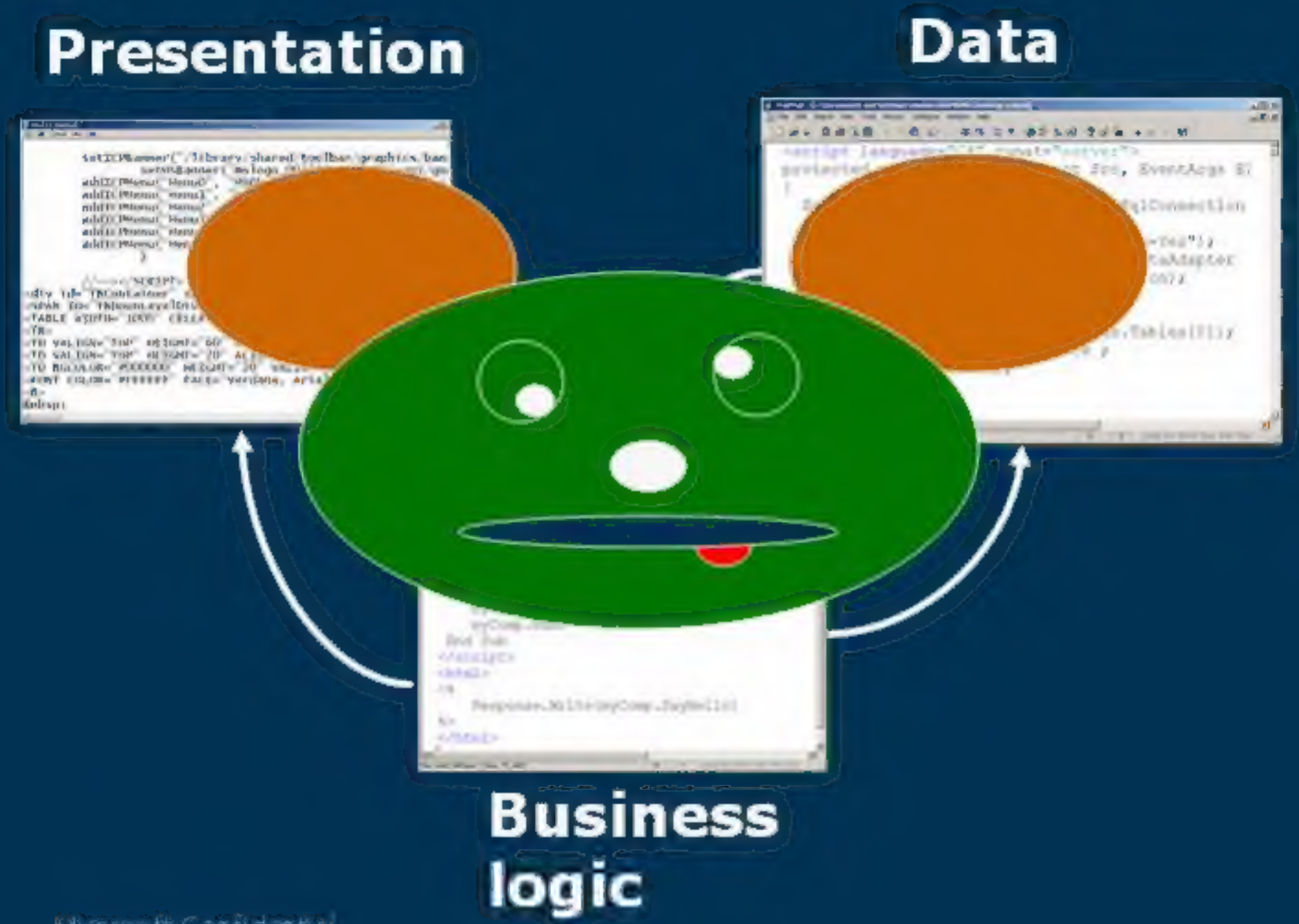
XEN: Unifying XML, SQL and CLR

Wolfram Schulte
Erik Meijer
and
WebData

MSR TechFest 2003

Programming a 3 tier architecture





Growing a Language

- ..., from now on, a main goal in designing a language should be to plan for growth.

- The language should start small, and the language must grow as the set of users grows.

Guy L. Steele Jr.

… so that we unify SQL, XML and CLR!

```
void Render(HtmlTextWriter output) { output.Add(
 <table>
   <tr><th>Product</th>
       <th>Quantity</th>
       <th>Price</th></tr>
       {select <tr><td> {p.ProductName} </td>
                   <td> {o.Quantity} </td>
                   <td> {o.Price} </td>
               </tr>
        from o in db.OrderDetails inner join
             p in db.Products on p.ProductID==o.ProductID
       }
  </table>);
}
```

The layers of XEN

Streams

- Similar to C# Iterators

```
//FromTo(1,5) = 1,2,3,4,5
IEnumerable<int>
int* FromTo(int start, int end) {
   for (int i = start; i<=end; i++)
      yield i;
}
```

Suspend and Return

```
foreach(int i in FromTo(1,5))
    Console.WriteLine(i);
```

Assign and Resume

Tuples

- Similar to anonymous structs

No object identity

```
[int Quot, int Rem] QuotRem(int x, in x) {
  return [Quot = x/y, Rem = x%y];
}

int q = QuotRem(47,11).Quot;
int r = QuotRem(47,11)[1];
```

Name Access

Positional Access

Database Table

Tuple

Stream

[string ID, string Name, string Company,
string Address, string, City,
int Zip, string Phone] *

ID	Name	Company	Address	City	Zip	Phone
MORGK	Alexander Feuer	Morgenstern Gesundkost	Heerstr. 22	Leipzig	04179	0342-023176
BLAUS	Hanna Moos	Blauer See Delikatessen	Forsterstr. 57	Mannheim	68306	0621-08460
OTTIK	Henriette Pfalzheim	Ottilies Käseladen	Mehrheimerstr. [illegible]	Köln	50739	0221-0644327
QUICK	Horst Kloss	QUICK-Stop	Taucherstraße 10	Cunewalde	01307	0372-035188
TOMSP	Karin Josephs	Toms Spezialitäten	Luisenstr. 48	Münster	44087	0251-031259
ALFKI	Maria Anders	Alfreds Futterkiste	Obere Str. 57	Berlin	12209	030-0074321
FRANK	Peter Franken	Frankenversand	Berliner Platz 43	München	80805	089-0877310
KOENE	Philip Cramer	Königlich Essen	Maubelstr. 90	Brandenburg	14776	0555-09876
LEHMS	Renate Messner	Lehmanns Marktstand	Magazinweg 7	Frankfurt a.M.	60528	069-0245984
WANDK	Rita Müller	Die Wandernde Kuh	Adenauerallee 900	Stuttgart	70563	0711-020361
DRACD	Sven Ottlieb	Drachenblut Delikatessen	Walserweg 21	Aachen	52066	0241-039123

Union

- Similar to type tagged unions

```
(int | string) Approx(object[] arr) {
    int i = arr.Length;
    return (i > TooBIG) ? "TooBIG" : i;
}

int k = (int) Approx(int xs);
bool? b = Approx(…).StartsWith("T");
```

Cast Expressions

Member Access

Content Classes

- Similar to XSDs

```
class Email {
    [
        [ DateTime Sent,
          string From,
          string To,
          string? Subject   ] Header,
        [ string P          ]* Body
    ];

    public static Email OOF() { }
}
```

Fields with and without names

Normal members

XSD

```
<xs:element name="EMail">
 <xs:complexType><xs:sequence>
  <xs:element name="Header">
   <xs:complexType><xs:sequence>
    <xs:element name="Sent" type="xs:dateTime" />
    <xs:element name="From" type="xs:string" />
    <xs:element name="To" type="xs:string" />
    <xs:element name="Subject" type="xs:string" minOccurs="1" />
   </xs:sequence></xs:complexType>
  </xs:element>
  <xs:element name="Body">
   <xs:complexType><xs:sequence maxOccurs="unbounded">
    <xs:element name="P" type="xs:string" />
   </xs:sequence></xs:complexType>
  </xs:element>
 </xs:sequence></xs:complexType>
</xs:element>
```

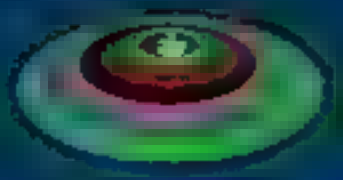
XML Literals

- Flexible object constructors

```
static Email OOF(DateTime s,
                 string t, DateTime u) {
  return <Email>
           <Header>
             <Sent>{s}</Sent>
             <To>{t}</To>
             <From>Wolfram </From>
           </Header>
           <Body>
             <P> OOF until {u}.</P>
             <P> Best, Wolfram </P>
           </Body>
         </Email>;
}
```

placeholders

Generalized Member Access



- Similar to XPath

```
//2/18/2003 "Erik" "Wolfram"
(DateTime | string)* info = oof.Header.*;
```

Wildcard

```
//"OOF"
string* subjects = oof..Subject;
```

Transitive

Generalized Member Access

```
//"OOF until 2/28/2003", "Best - Wolfram"
string* ps = oof.Body.P;
                    [string P]*
```

apply member access to all

```
// "Erik", "Wolfram"
string* ToFrom = oof.Header.string:*;
```

type-based access

Join

SQL's select-from-where

```
[DateTime Sent, string? Subject]*
  msgsToSelf =
    select i.Sent, j.Subject
    from i in inbox, j in inbox
    where i.To == j.From;
```

SVG Demo

- Scaleable Vector Graphics (SVG) is an XML grammar for graphics
- SVG content can be displayed in Web pages
- Use XEN to generate an SVG form

Typeful Programming

- Types are essential for the ordered *evolution* of large software systems.

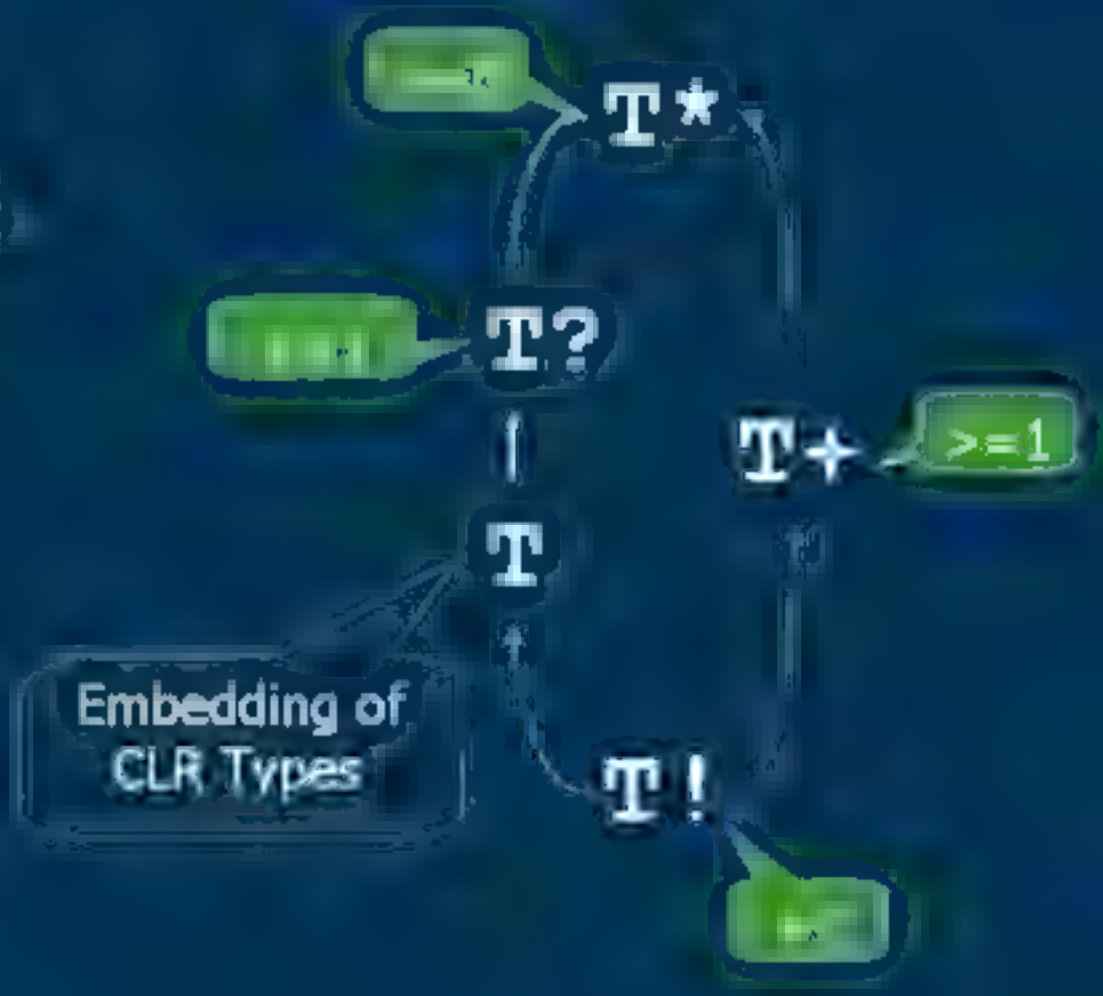
- Subtypes are essential for the ordered *extension* of large software systems.

Lucca Cardelli

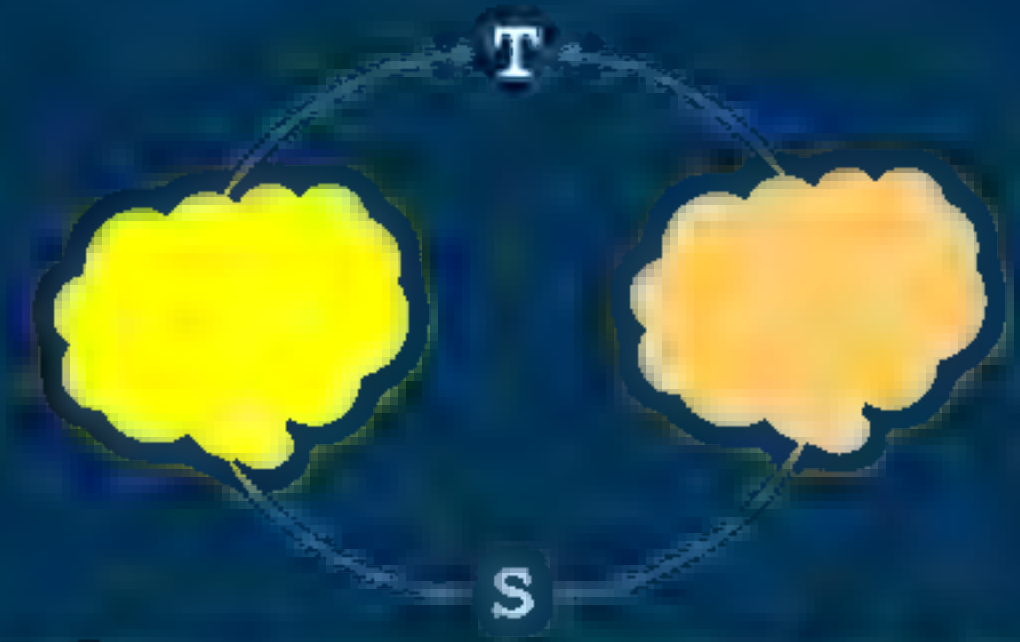
Stream Subtyping (simplified)

- Similar to language (stream) inclusion
- Null = empty stream
- Streams are covariant

$$\frac{S <: T}{S^* <: T^*}$$

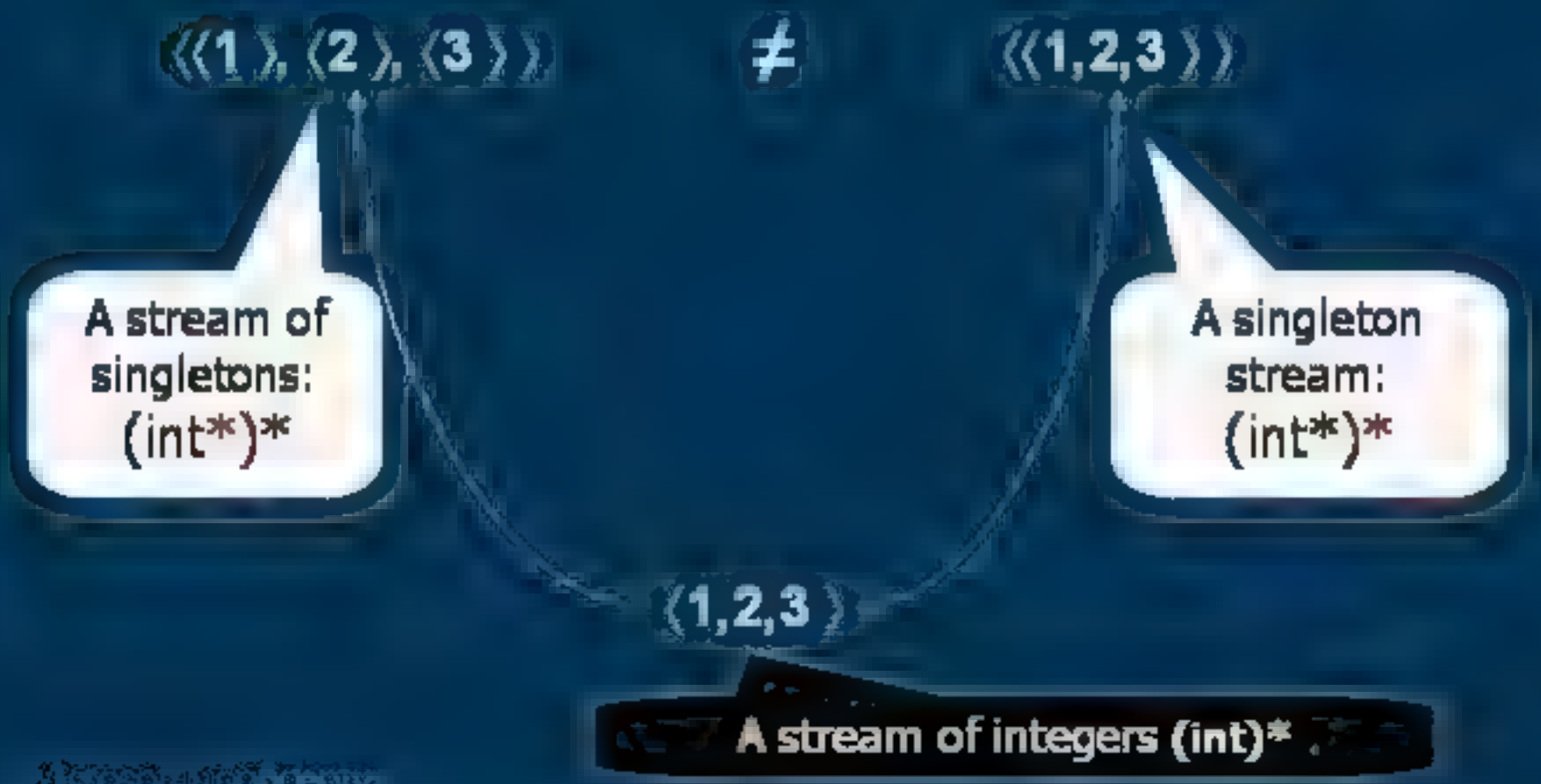


Problem of Subtyping: Coherence



Different ways to upcast from S to T must result in an equivalent value

Stream Subtyping: Possible Loss of Coherence



Stream Subtyping: Flattening to Preserve Coherence

- T* = T**
- T+ = T++
-

The Path to XEN

Conclusion

- XEN unifies CLR, SQL and XML.
- XEN makes you more effective
- XEN programs are more trustworthy
- XEN build is available via http://xsharp
- XEN release: April 2003

Thank you